RECEIVED
SEP 12 1997
FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

Outside Plant (Loop) Design

by

W. W. Sharkey, D. M. Kennet, & C. A. Bush,

Federal Communications Commission (FCC)

&

Vaikunth Gupta,

Panum Telecom, LLC

(301) 299-6271

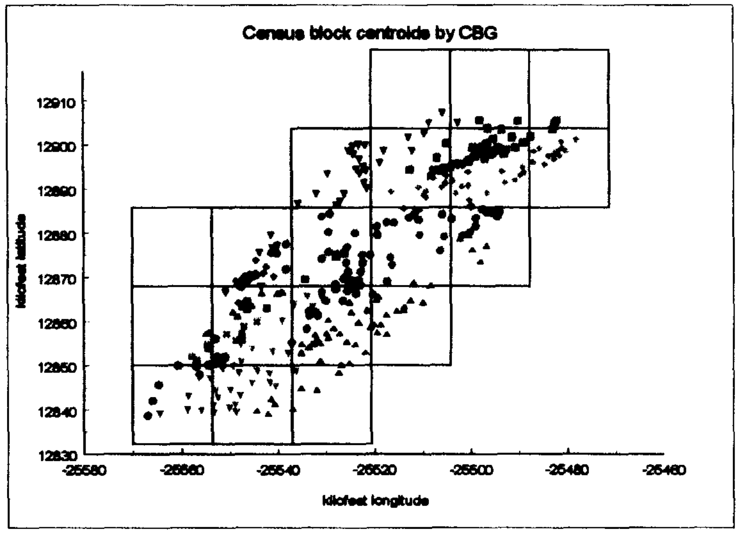
Sept. 11, 1997

Loop Design - Objectives

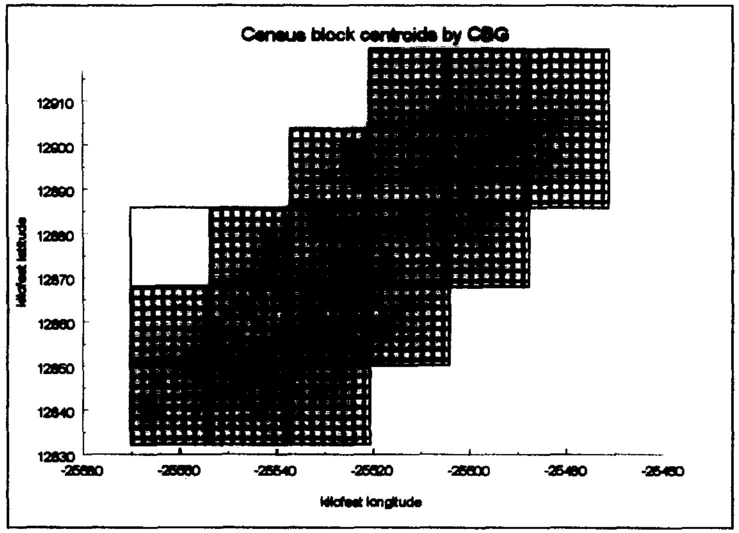
- **Use FCC Staff's Customer Location Design Methodology to Derive**
  - Optimized Cost Estimates for
    - Efficient Distribution Plant,
    - Sub-feeder Layout, and
    - Feeder Layout
  - Loop Design based on Sound Engineering Principles
  - Uncompromized Network Performance for Narrow band POTS Services

Customer Location Overview

- **Raw Data**



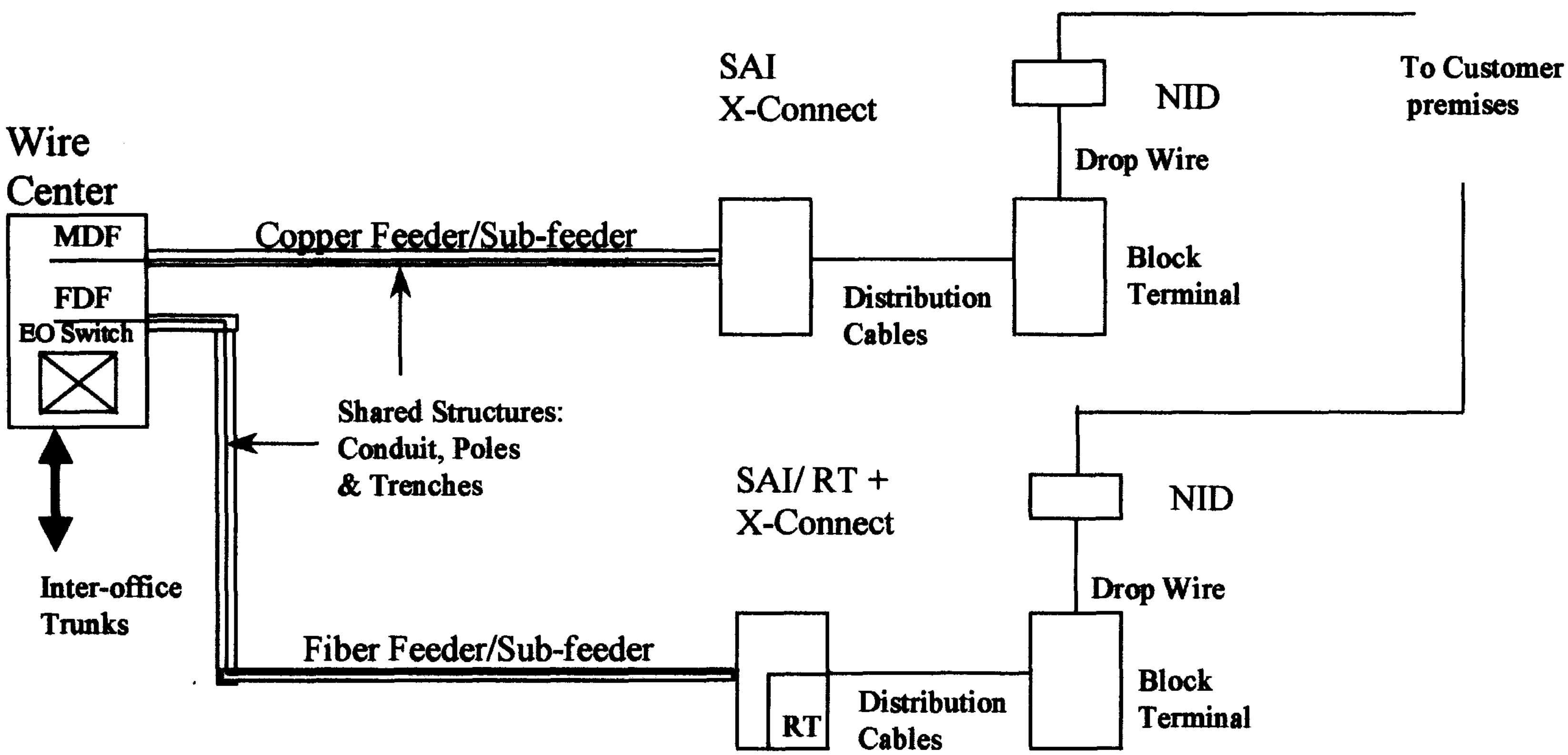
- **Grids and Micro-grids**



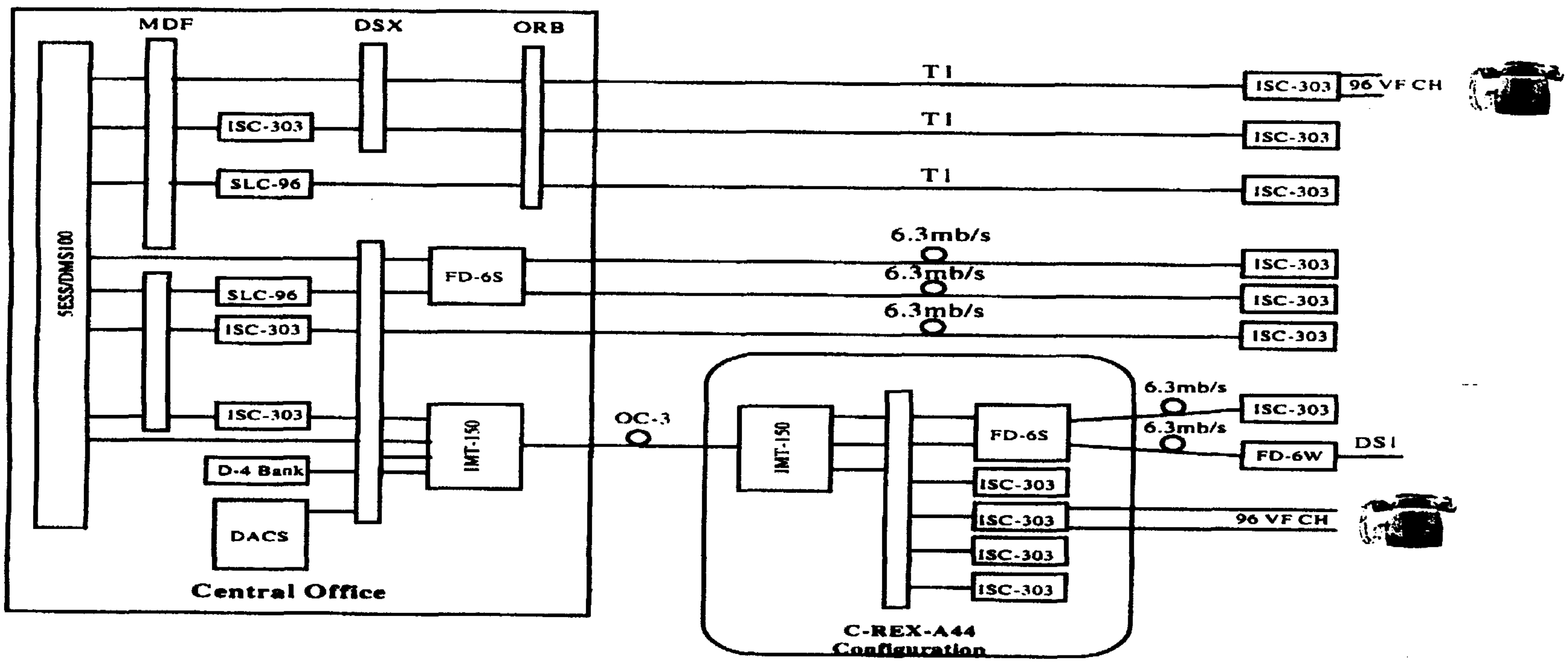
HCPM- Loop Design:
Draft Interim Progress Notes

All data used for illustration purposes only

Network Topology & Terminology

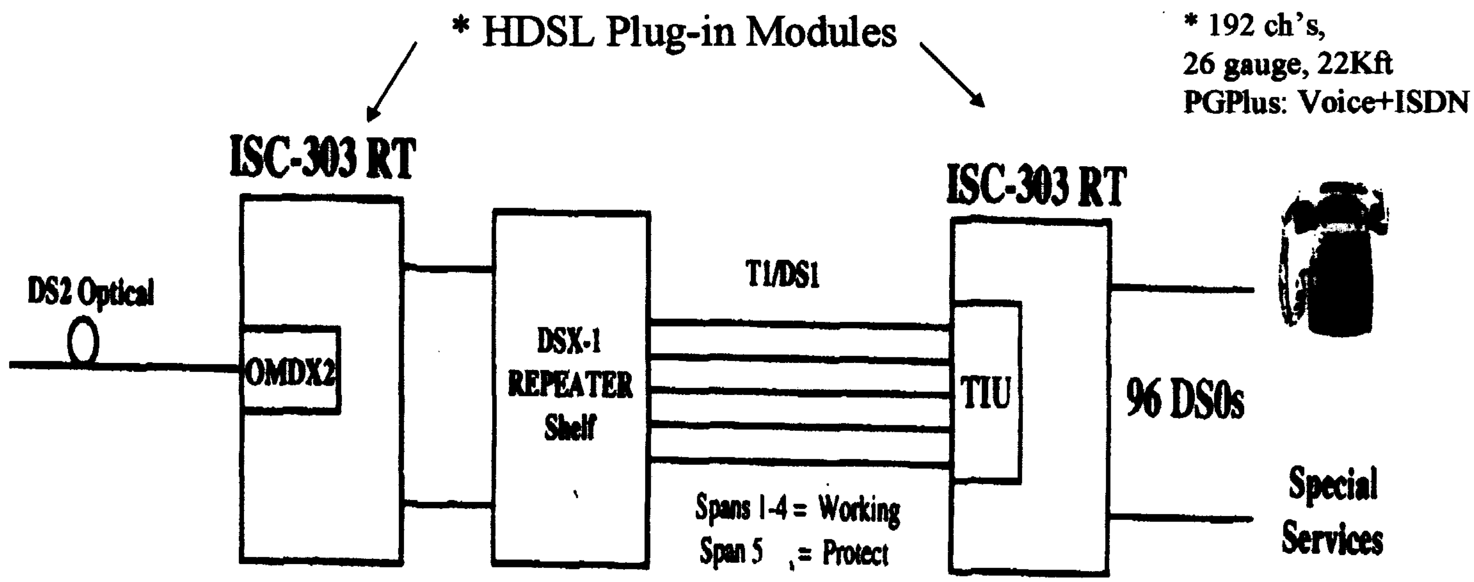


Network Topology & Terminology (Continued)



Ref: NEC 's ISC 303 Planning Guide, Issue 1, Page 22

Network Topology & Terminology (Continued)



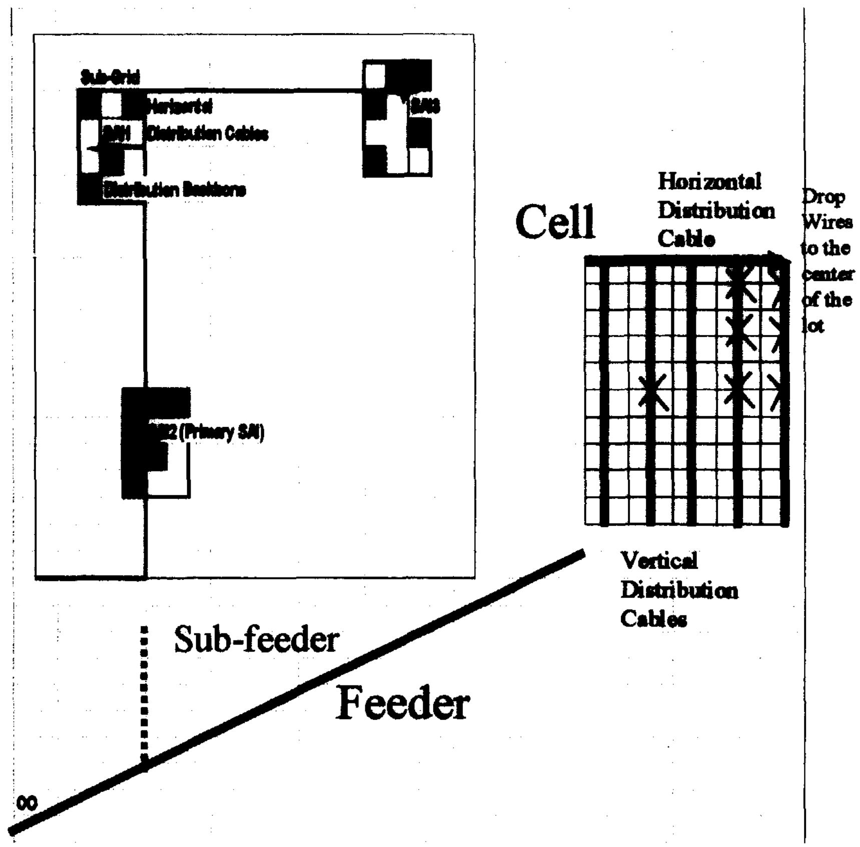
Ref: NEC 's ISC 303 Planning Guide, Issue 1, Pages 24 & 26
* HDSL Data: Conversations with Pairgain Technologies, 75% market share of HDSL in the US

Loop Design Highlights

- Optimal Use of Fiber DLC, T1 on Copper and Fiber/Copper Cables
- Use of Copper T1 Technology on Long Loops in Thinly Populated Areas
- Fine Granularity in the Use of DLC Terminals
- Multiple SAIs Located in Each Grid
- Unique Implementation of a Hybrid between Hatfield & BCPM
- Use of Flexible Data Structures and High Level Language for Adequate Optimization

Loop Design - Overview

- **Customer Location Module provides coordinates and potential SAI locations.**
- **Grid (18 kft square or less) Contains Cells (Micro-grids) with non-zero population/# of lines estimates**
- **Design Distribution plant for each grid in isolation based on on cost minimization.**
- **In each quadrant, compute optimal angular feeder routes using locations of primary SAIs.**



Rules: Feeder/Subfeeder/Distribution

1. **Copper cables used if**

   (Max Copr Distance from CO < Max Copr Distance Thresh) and

   (# of lines in Grid < Min Grid Lines served by T1 Copr Technology Thresh)

2. **T1 Copr technology is used starting at the CO if**

   (Feeder Distance < FibCoprThresh) and

   (Min Grid Lines served by T1 Copr Technology Thresh <= # of lines in Grid < Min Grid Lines served by Fiber Technology Thresh)

3. **Use fiber from CO If**

   (Feeder Distance >= FibCoprThresh)

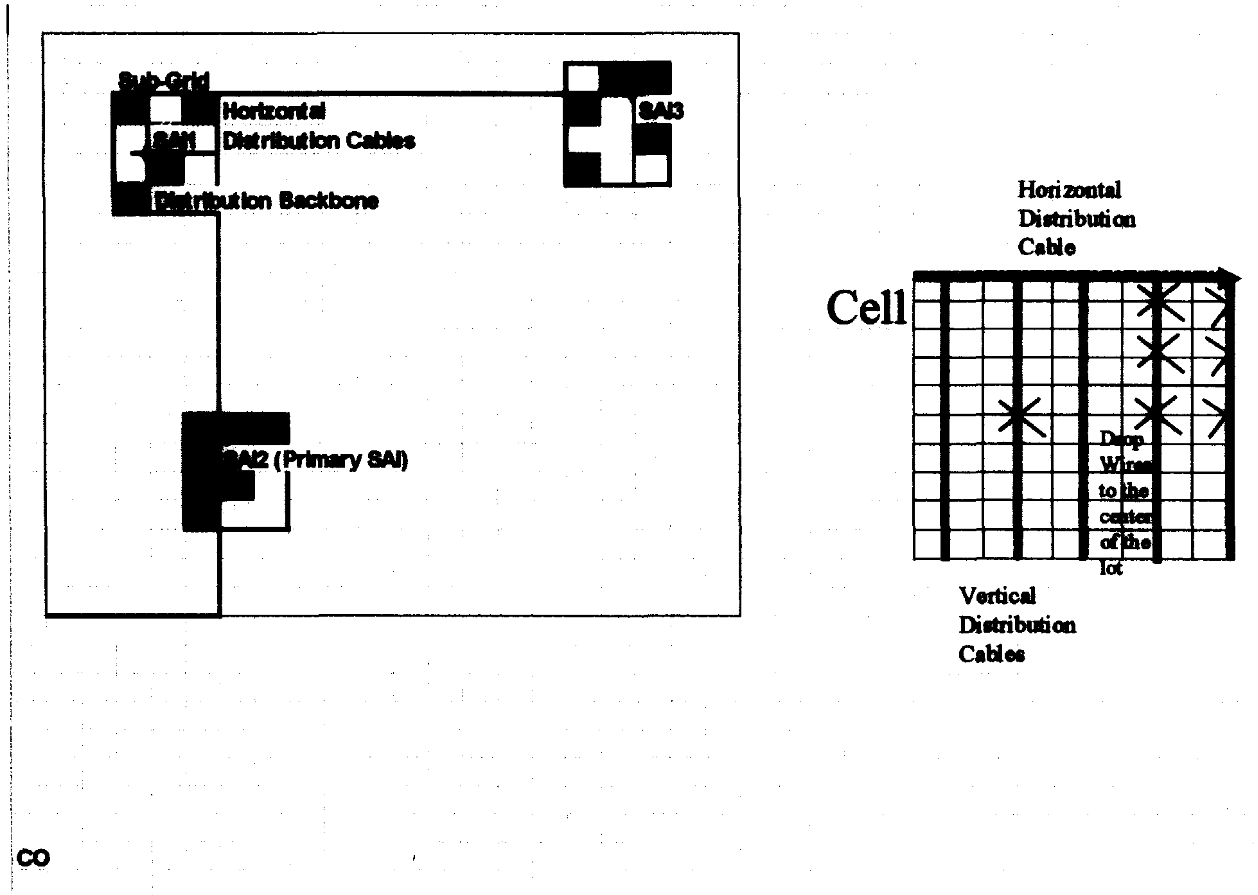
   or

   (# of lines in Grid >= Min Grid Lines served by Fiber Technology Thresh)

Distribution Design Algorithm: Step 1

- **Design Distribution Plant for a fixed set of SAIs**
  - Define SA boundaries (Sub-grid) by attaching each cell (micro-grid) to the nearest SAI using the L1 distance
  - Divide each cell into lots by dividing area by population (residential + business); residence/business is assumed to be located at the lot center
  - Place Distribution backbones to touch every lot within a cell and connect; identify a unique corner of each cell that is closest to the SAI such that distribution cable can be placed on every other cell boundary
  - Connect each cell to its SAI by a shortest L1 path and size cables appropriately on the path back to the SAI

Distribution Architecture



Distribution Design Algorithm: Step 2

- **Optimize the number and location of SAIs in each grid**
  - Determine the SAI closest to the CO in L1 distance (Primary SAI), determine if this location may be served by analog cables, T1 copper or fiber DLC, and compute provisional feeder cost based on L1 distance.
  - Determine the size of the DLC or T1 Copper Terminal(s) at each of the other Secondary SAIs and using the minimum L1 distance from the Primary SAI determine whether this distance needs to be served using copper or fiber.
  - Resize the Primary SAI.
  - Also compute provisional feeder costs assuming some of the SAIs are directly connected to the provisional feeder entry point into the grid.
  - Compute the cost for the final configuration for each combination of Primary and Secondary SAIs and select the combination with the least cost.
  - Disregard the provisional feeder costs and store the SAI configurations and distribution costs for future reference.

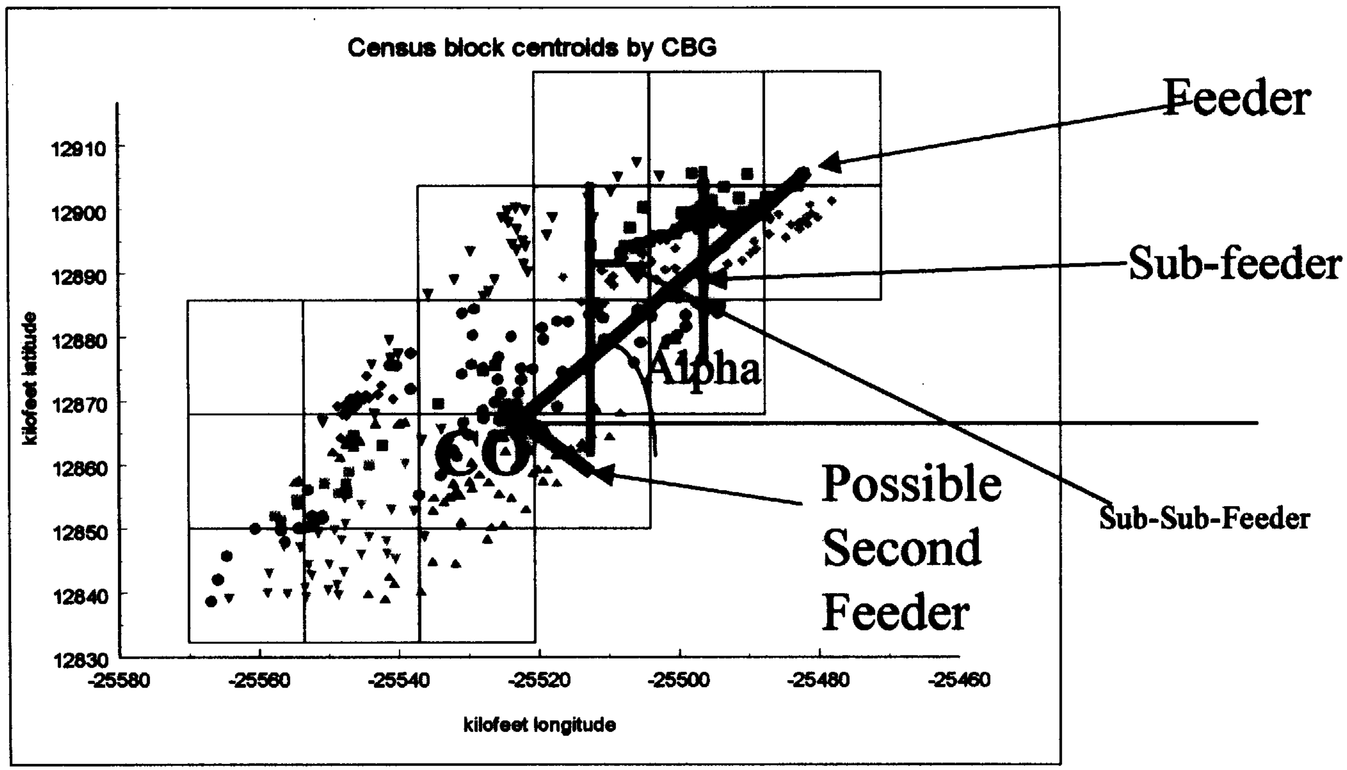
Feeder Design - Single Feeder:

- **Compute the angle defined by the average of all Primary SAIs in a quadrant**
  - α =ArcTan[Weighted sum of y-coordinates of all SAIs/Weighted sum of x-coordinates of all SAIs]
- **For each vertical block of grids, compute the average x-coordinate for all SAIs (X^j). The Sub-feeder will run along this vertical line through X^j. The Primary SAI in each grid will be connected to this sub-feeder.**
- **Compute the total feeder + sub-feeder distance and the associated cost for the quadrant based on cable size and structure cost (A similar procedure applies to horizontal blocks of grids when appropriate).**

Feeder Design - Dual Feeders:

- **Define α_1 and α_2 for the upper and lower half of each quadrant. Use the same method as for a single feeder except sum over the SAIs within the upper and lower halves of the quadrant.**
- **Determine if the sub-feeder is vertical or horizontal and compute distance for the sub-feeder.**
- **Determine feeder + sub-feeder cost for the quadrant.**
- **Use the lower of the cost for the single feeder or the dual feeder implementation to determine the final feeder cost.**
- **Re-compute the true cost for each grid using the feeder/sub-feeder cost data.**

Feeder/Sub-feeder Architecture



All data used for illustration purposes only

Attachments

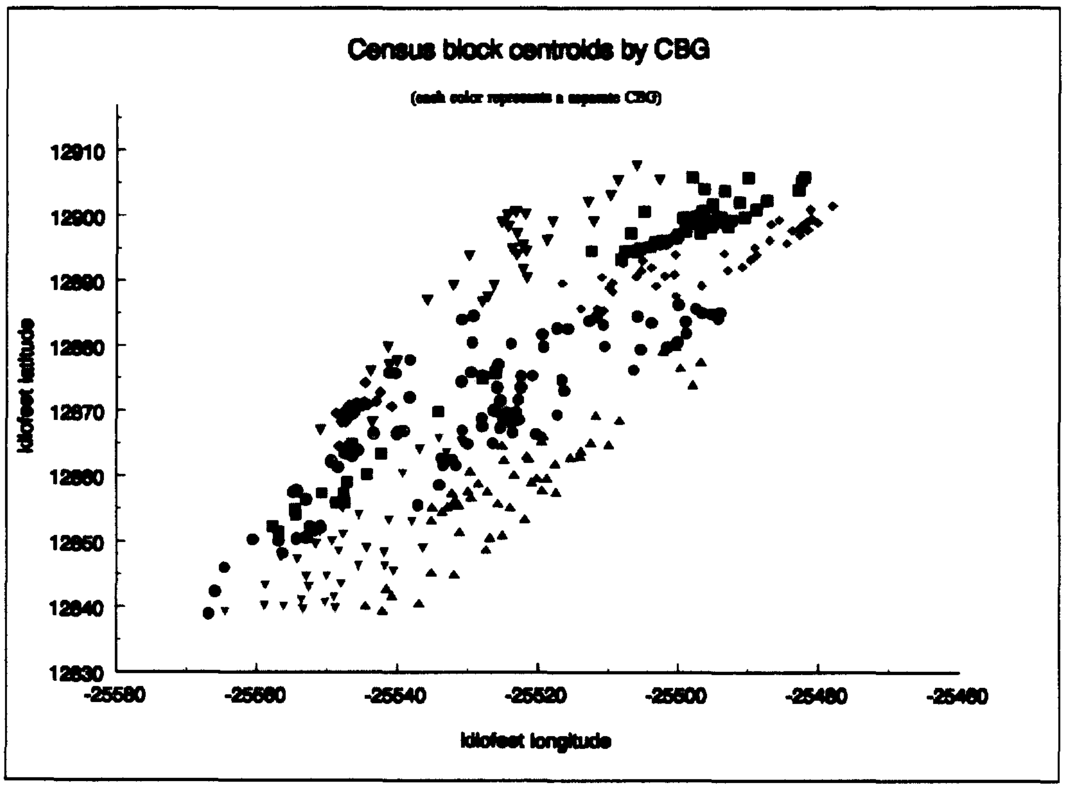
- **Customer Location Overview**
- **Loop Design Algorithms**
- **Psuedocode for costing the Distribution plant and locating the SAIs**
- **Psuedocode for costing the Feeder/Sub-feeder plant and updating the Grid Costs**
- **Resume for Vaikunth Gupta**
- **Company Profile for Panum Telecom, LLC**

AN APPROACH TO MODELLING CUSTOMER LOCATION

D. Mark Kennet
Federal Communications Commission

In this short paper, I attempt to explain an approach to the problem of modelling customer location for forward-looking economic cost models expected to be employed in universal service fund (USF) calculations as well as unbundled network element (UNE) proceedings. This approach has been incorporated into a software product I call CENBLOCK, and is best explained using an example.

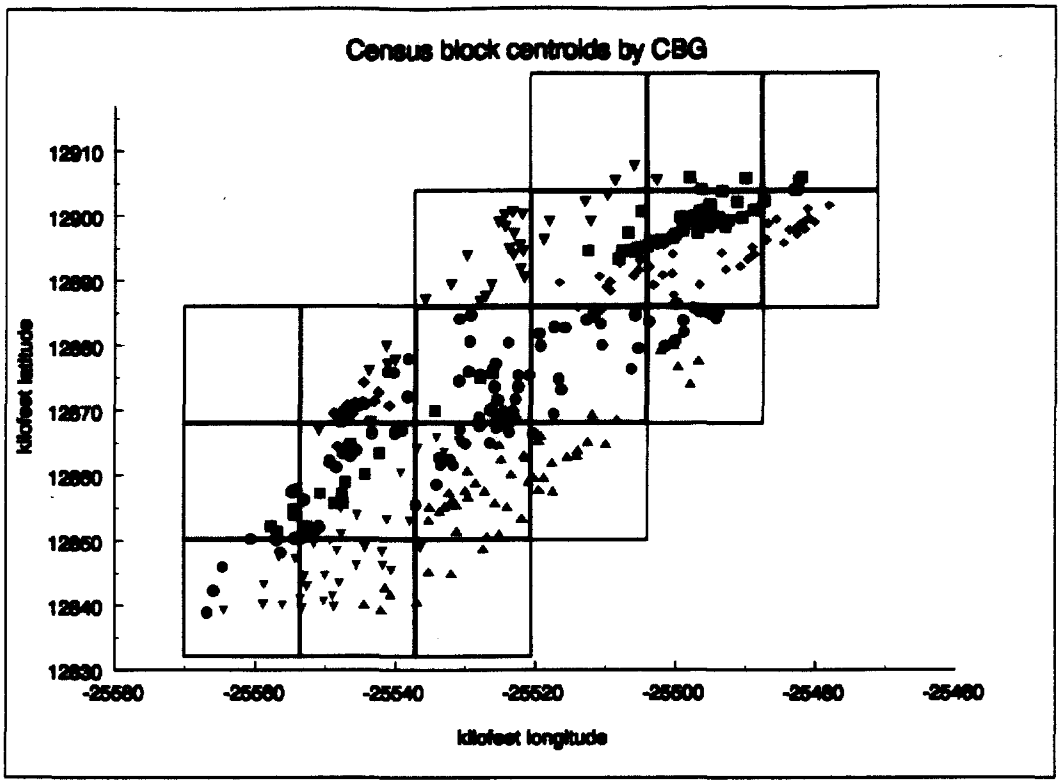
The drawings below illustrate a dataset consisting of the Census blocks within a particular telco's service territory. In the first drawing, I have plotted the raw data. Each point on the map represents the centroid of a block.



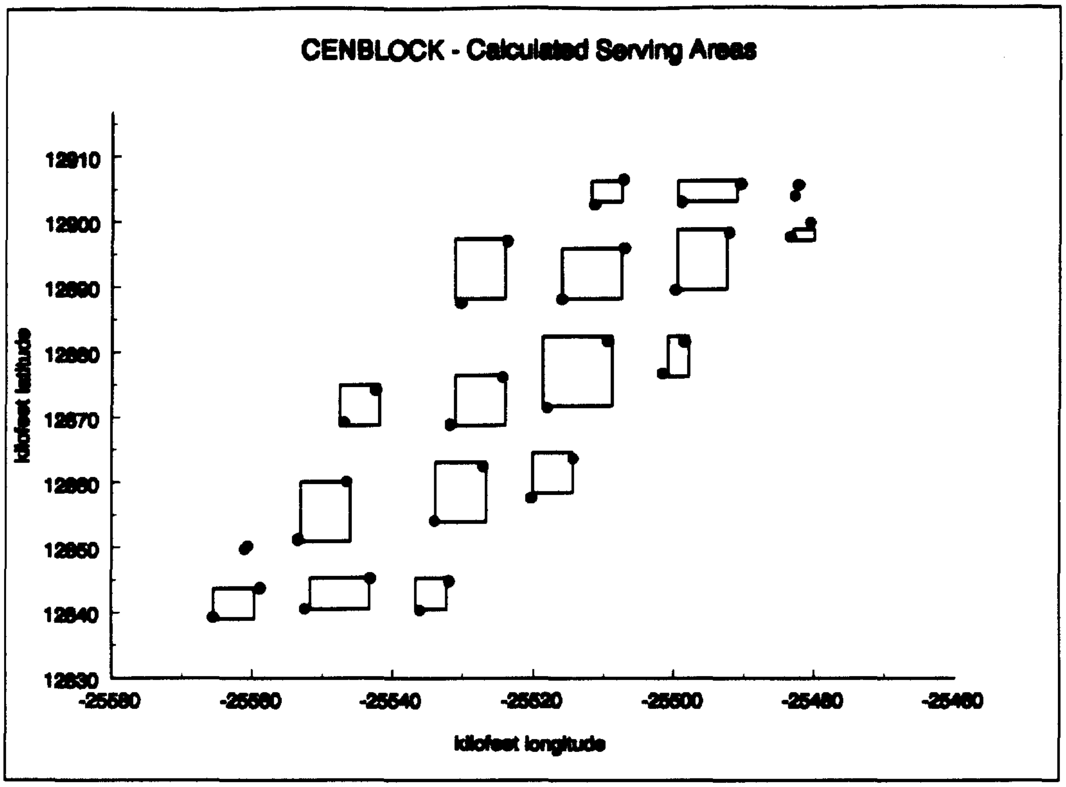
In modelling the feeder and distribution plant for a telco, if all we had available were the above data, we would need to assume some aggregation rule for the Census blocks. If each of the squares (small, medium, large) was a serving area, with its own serving area interface and distribution plant, the model would clearly overstate the amount of serving area interface electronics. On the other hand, aggregating to the level of the Census block group (CBG) level will misrepresent what is evident in this rural dataset (taken from a rural section of Tennessee): some CBGs have population located outside the reach of any currently existing interface devices, even if the area of the CBG meets an engineering specification. I would argue that in the context of engineering telco plant, length and width of distribution areas are more important than the area, for exactly this reason. The other point worth noting is that there is significant clustering of population, even in this rural section.

The CENBLOCK software provides a means to systematize that clustering for modelling purposes, aggregating together the blocks so that more meaningful approximations to real-world serving areas can be used while preserving information about length and width of the clusters.

The simplest clustering arrangement would be if we assumed that all serving areas were squares. The picture below depicts a CENBLOCK arrangement of 18 kf by 18 kf squares, with a maximum population of 2000 customers per serving area (the latter constraint in this dataset is nonbinding). If any of the grid blocks were to exceed that population count (which is a user input), CENBLOCK repeatedly cuts the block in half until either the maximum population criterion is met or it is determined that only one Census block remains in the grid (and no further divisions are possible).



Note that in this drawing, the total number of serving areas is equal to the number of populated squares, in line with what would likely be observed in the actual feeder/distribution plant, but the shape of the distribution areas does not adequately address clustering within the grid block. To address the latter issue, I make the following initial proposal for an algorithm to model the distribution area boundaries. First, find the centroid of all included CBs in a grid block. Now calculate the average X-value for CBs to the left of that centroid, and that for CBs to the right. Repeat the exercise for Y-values. There are now four corners, representing an "average" both horizontally and vertically for the locations of customers in the stylized serving area. For this example, modelled serving areas are as follows:



One may argue that these stylized distribution areas also do not quite capture population clustering or distances quite adequately. To address this problem, I have devised an alternative approach. I create "microgrids" within each grid block, with the size equal to the average area of Census blocks in that grid. This approach enables locating customers with the finest level of resolution supported by the data.

My approach is best understood through a pseudocode description of the process:

PSEUDO-CODE FOR CENBLOCK PROGRAM

`get_parameters;`

> *Loads user-defined values for parameters, including size of grid, maximum allowable population within a distribution area, size of microgrid (if user desires to preset this), and take rate for households.*

`setup_grid;`

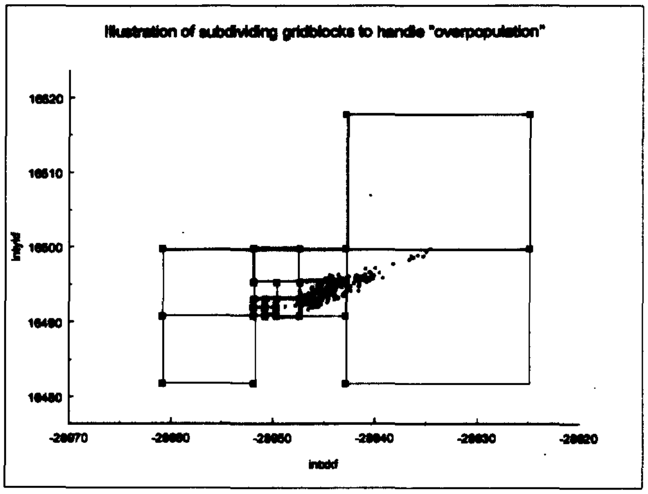
> *Using data from included Census blocks, calculates upper left and lower right corners of area to be "gridded." Writes corners of grid blocks to file.*

attach_mesh_areas;

Loops through each grid block created above, and attaches all Census blocks that fall within that grid by saving "addresses" in a file. Also tabulates the number of lines within each grid block and saves info to file.

while **over_max_pop** do **adjust_grid_areas;**

Over_max_pop is a Boolean function that returns "TRUE" if any grid block has a number of lines over the maximum defined in the parameter file. If any such grid blocks exist, adjust_grid_areas divides them into smaller square grids and determines which Census blocks are attached to the smaller grids.



build_serving_areas;

This module accomplishes two tasks before writing distribution area data to file. First, it grids each grid block to determine the location of the Census blocks contained within it. If this microgrid is preset by the user, it will use that value; otherwise, it will determine the average area of included Census blocks to determine a microgrid size. Here is an illustration:

number of rows, cols: 22,22
Lower Left X,Y: -28832.7347 16286.8341
NS Grid size: 0.8182
EW Grid size: 0.8182
Centroid: -28827.8256,16290.9250
Household matrix:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	49	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	13	0	53	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	13	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	12	18	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	11	25	61	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	16*	51	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	30	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Second, it optimizes the location of each serving area interface by minimizing a "cost function" that assigns a penalty to any SAI location according to the sums of absolute rectilinear distances from each microgrid midpoint times the number of lines in that microgrid:

$$Cost(x,y) = \sum_i \sum_j \left[abs(midx_{ij} - x) + abs(midy_{ij} - y) \right] pop_{ij}$$

It now asks the question, "if we were to put more than one SAI in this gridblock, where would they go?" It then optimizes the locations of up to four SAIs within each gridblock. Finally, the data are output in a format consistent with BCPM (it is a trivial matter to output to a Hatfield format), with an auxiliary binary file containing the microgrid information.

Hybrid Cost Proxy Model
Loop Design Algorithms

Preliminary and Incomplete:
The following notes do not address all relevant loop design issues and are subject to modification.

0. Setup

0.1 Notation

For expositional purposes in these notes, grids will be identified by indices i and j representing the displacement from a bottom-left corner of the smallest rectangle of grids which contain a wire center. Grids are indexed G_{ij} for i,j = 1 , 2 , ... , and it is assumed that the user adjustable grid size is set at 18 kilofeet throughout these notes. Each grid is subdivided into an m by m matrix of micro grids (also called cells), where m ranges between 1 and 50. A serving area interface contained in G_{ij} is identified by SAI_{ijk}.

Distribution consists of all outside plant between a primary serving area interface and the customer location, and may include secondary SAIs. Distribution plant consists of backbone and branching cable where branch cable is closer to the customer location.

Feeder consists of all outside plant between the central office main distribution frame or fiber distribution frame up to and including a primary serving area interface. Feeder cable consists of main feeder, sub-feeder and sub-sub-feeder routes.

0.2 Inputs from the Customer Location Module

The Customer Location Module will report for each grid the bottom-left and top-right coordinates of the grid, the number of cells, the number of lines associated with each cell, and an array defining the locational coordinates for four possible configurations (consisting of 1, 2, 3, and 4 locations) of SAIs to serve the grid.

0.3 Rules for Feeder, Subfeeder and Distribution Plant

Copper cable is used if the maximum copper distance from the CO is less than the maximum copper distance threshold and the number of lines in the grid is less than the T1 copper technology threshold.

T1 copper technology is used starting at the CO if the feeder distance is less than the fiber copper threshold and the number of lines in the grid is greater than or equal to the T1 copper technology threshold but less than the fiber technology threshold.

Fiber technology is used starting at the CO if the feeder distance is greater than or equal to the fiber copper threshold or the number of lines in the grid is greater than or equal to the fiber technology threshold.

0.4 Cable and Structure Costs

For a given technology t (e.g. 26 gauge analog copper, T1 on copper, etc.) costs are assumed to be of the form $(a_t + b_t L) D$, where L represents the number of lines and D represents distance in kilofeet. The fixed cost per kilofoot, a_t represents the cost of structures and is determined from input tables. The variable cost b_t is also determined from input tables.

1. Distribution Plant

Design the distribution plant for each grid in isolation (ignoring information from all neighboring grids). The end result of this stage is a set of one to four primary SAI locations within each grid which will remain fixed throughout all future steps. Associated with each primary SAI_{ijk} will be a vector (x_{ijk} , y_{ijk} , L_{ijk} , $DistributionCost_{ijk}$) defining the horizontal and vertical coordinates, the number of associated lines, and the total distribution cost of all plant served by that SAI, including the cost of any associated secondary SAIs.

1.1 Divide each cell into lots based on cell population. Identify the corner of each cell that is closest to its serving SAI and which lies on an even horizontal or vertical branch route. Place distribution branch cable to touch every lot within a cell and connect to this corner point. Design drop cable to serve groups of four properties whenever possible.

1.1.1 Cell dimensions are (18/m) kilofeet on a side. Suppose that L residential lines and no business lines are required in the cell. The cell is divided into $n_1 * n_2$ lots where $n_1 = \textit{UpperInteger}(\sqrt{L})$ and n_2 is the smallest integer such that $n_1 * n_2 \geq L$. Lot dimensions are therefore $18/(m*n_1)$ by $18/(m*n_2)$.

1.1.2 Backbone and branching distribution cable is built to touch every lot in the cell. From any corner of the cell, the backbone follows the side containing n_1 lots and is of length $\frac{18(n_1 - 1)}{m * n_1}$. Branching cable is required on every other lot boundary. Therefore *UpperInteger*$(n_1/2)$ branching cables are required. In order to touch every lot, each branch must be of length $\frac{18(n_2-1)}{m * n_2}$. The total cost of backbone and branching cable in a cell using technology t is therefore given by

$$\frac{18}{m}\left[\left(\frac{n_1-1}{n_1}\right)+UpperInteger\left(\frac{n_1}{2}\right)\left(\frac{n_2-1}{n_2}\right)\right].$$

Apply cable sizing and structure cost subroutines to obtain backbone and branching costs.

1.1.3 Drop length is $\sqrt{\left(\frac{18}{2*n_1}\right)^2+\left(\frac{18}{2*n_2}\right)^2}$. If drop cost per kilofoot is equal to d, total drop cost in the cell is given by

$$L*d*\left(\sqrt{\left(\frac{18}{2*n_1}\right)^2+\left(\frac{18}{2*n_2}\right)^2}\right).$$

1.2 For each possible configuration of SAIs in a grid, the serving area for SAI_{ijk} is defined as the set of cells closer to it than to any other SAI in the grid in terms of L1 distance. The smallest rectangle containing a serving area is called a *subgrid.*

1.2.1 Define the bottom-left and top-right coordinates of a subgrid serving SAI_{ijk} respectively as $(\underline{x}_{ijk}, \underline{y}_{ijk})$ and $(\hat{x}_{ijk}, \hat{y}_{ijk})$. Main distribution backbone from (x_{ijk}, y_{ijk}) will be horizontal if

$$\text{Max}\left(x_{ijk} - \hat{x}_{ijk},\ \underline{x}_{ijk} - x_{ijk}\right) > \text{Max}\left(y_{ijk} - \hat{y}_{ijk},\ \underline{y}_{ijk} - y_{ijk}\right)$$

and vertical otherwise. (This will minimize the combined length of the backbone and branching plant and therefore minimize structure costs.) Compute the L1 distance from (x_{ijk}, y_{ijk}) to each populated cell in the subgrid assuming only even cell dividing lines are used for distribution branches.

1.2.2 If the backbone is horizontal it will be of length Max $\{x_1, \hat{x}_{ijk}\}$ - Min $\{x_1, \underline{x}_{ijk}\}$ - 18. The number of branching cables